

Computer Science at Oxford

Michael Spivey

Tutor, Oriel College



UNIVERSITY OF
OXFORD

Department of
COMPUTER
SCIENCE

Copyright © 2013 J. M. Spivey

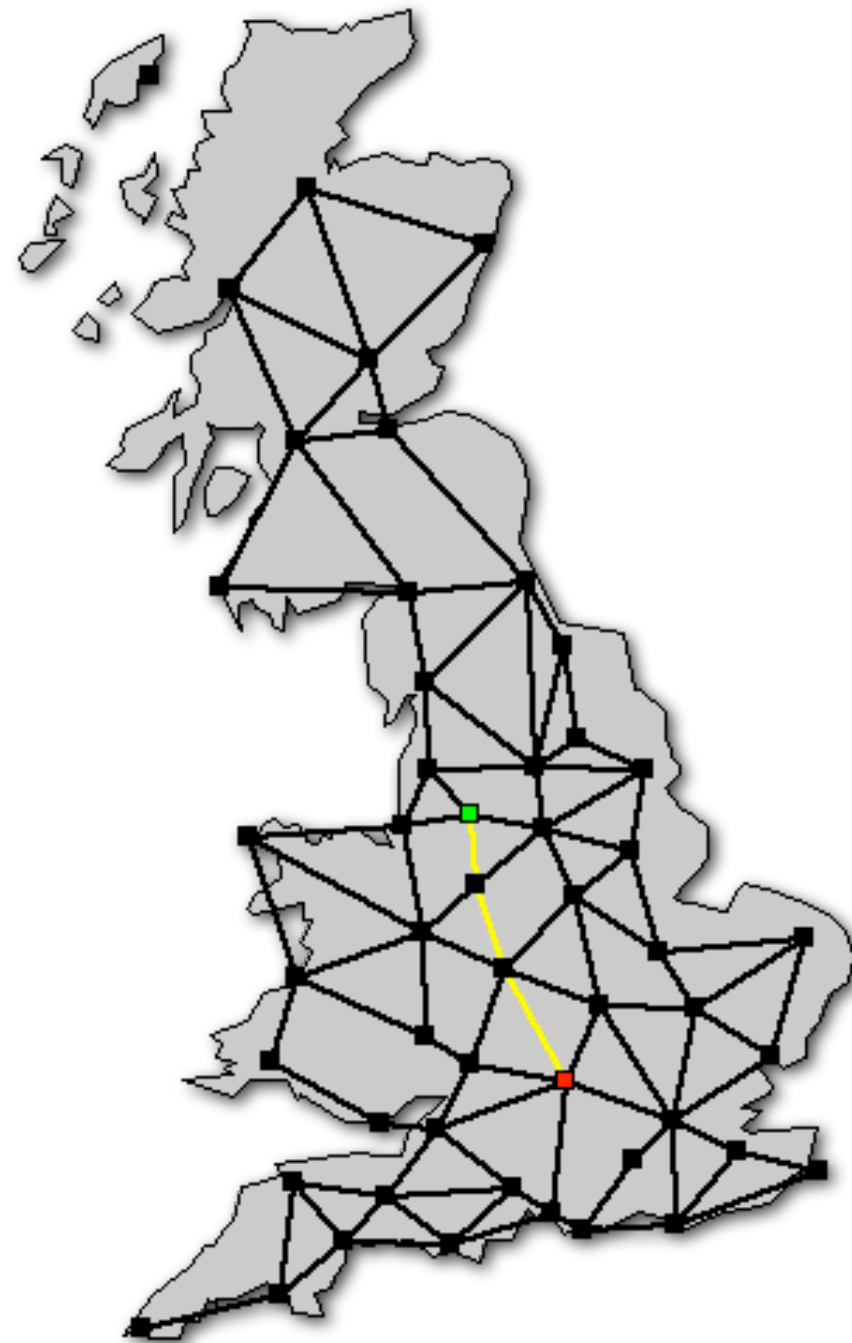
Computer Science at Oxford

- What's Computer Science about?
- The Oxford courses
- Four *myths* about Oxford

What's Computer Science about?

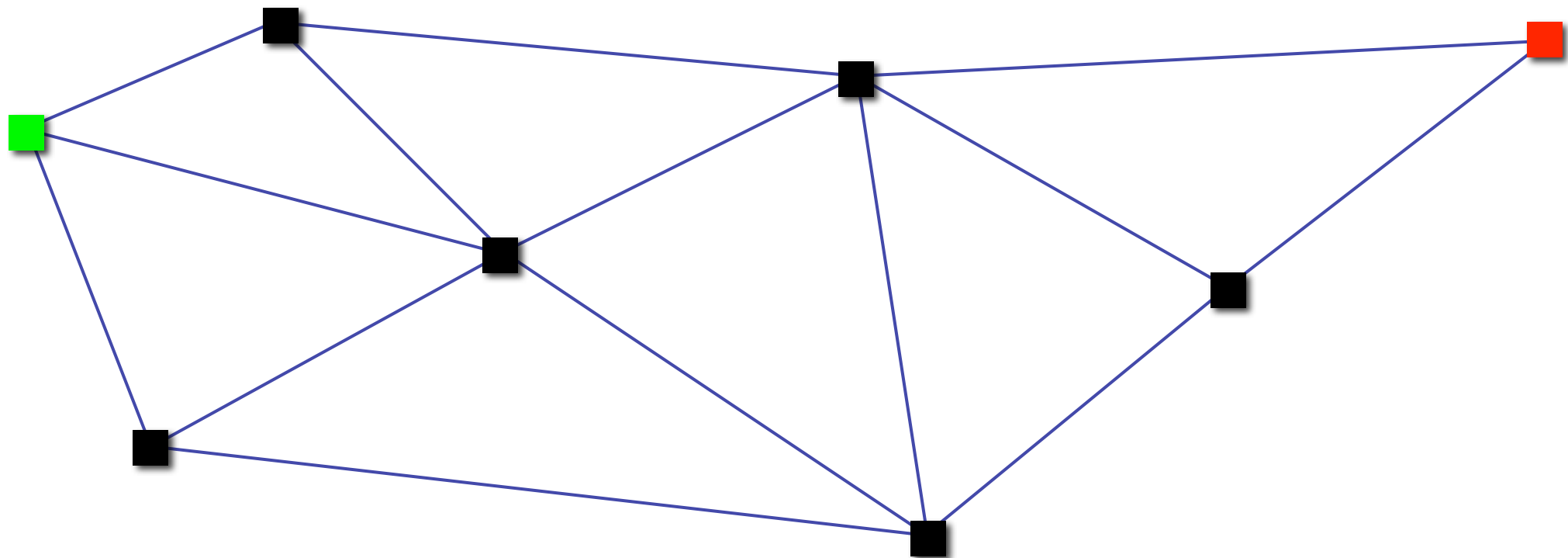
Or, how can you drive
from Manchester to
Oxford?

And how can you get a
computer to show you
the way?



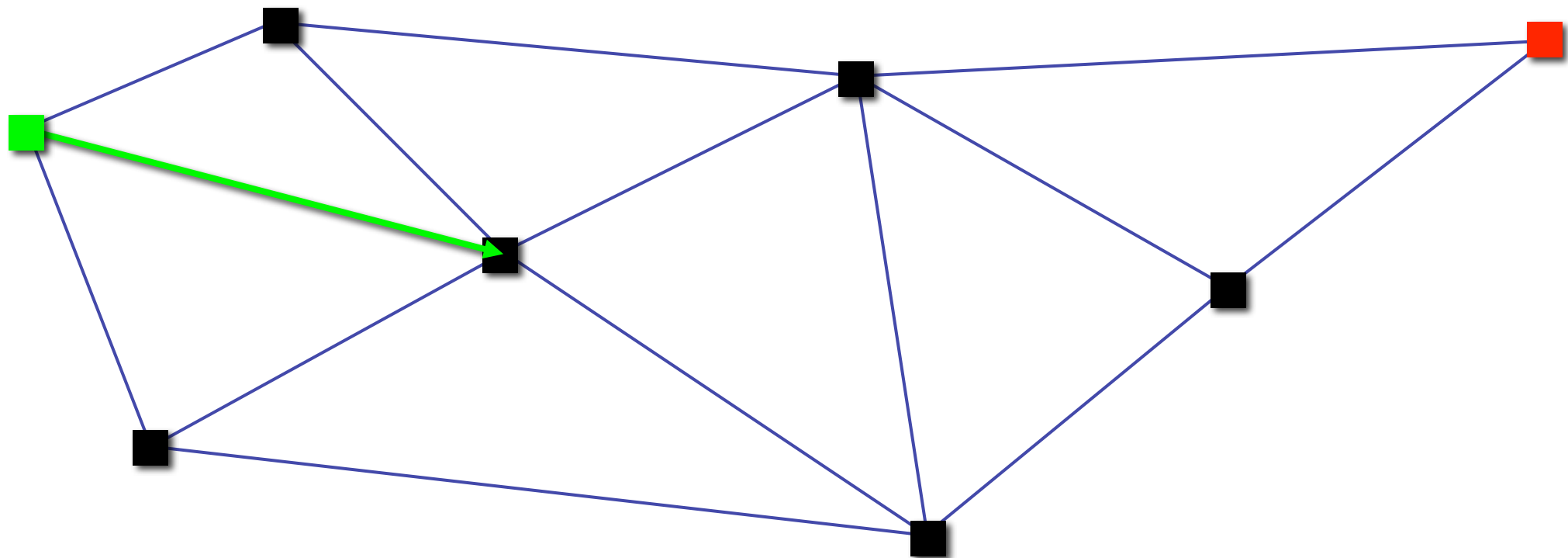
Method A

“Always go in the direction that takes you most directly towards Oxford.”



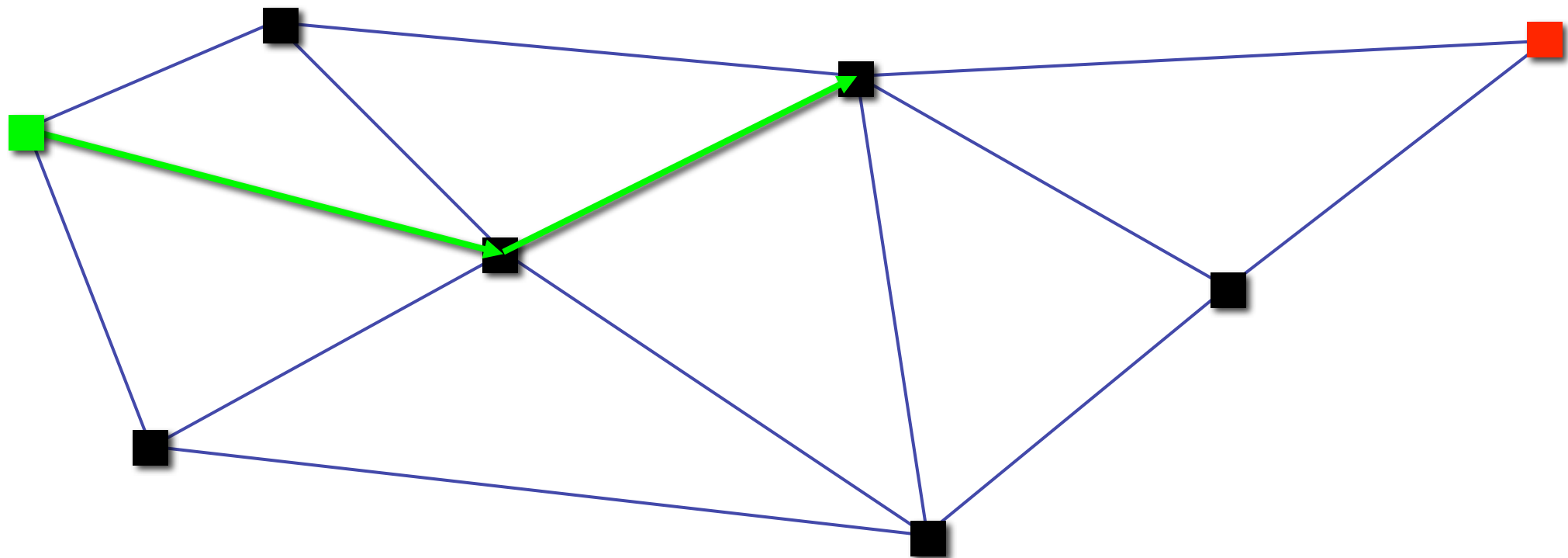
Method A

“Always go in the direction that takes you most directly towards Oxford.”



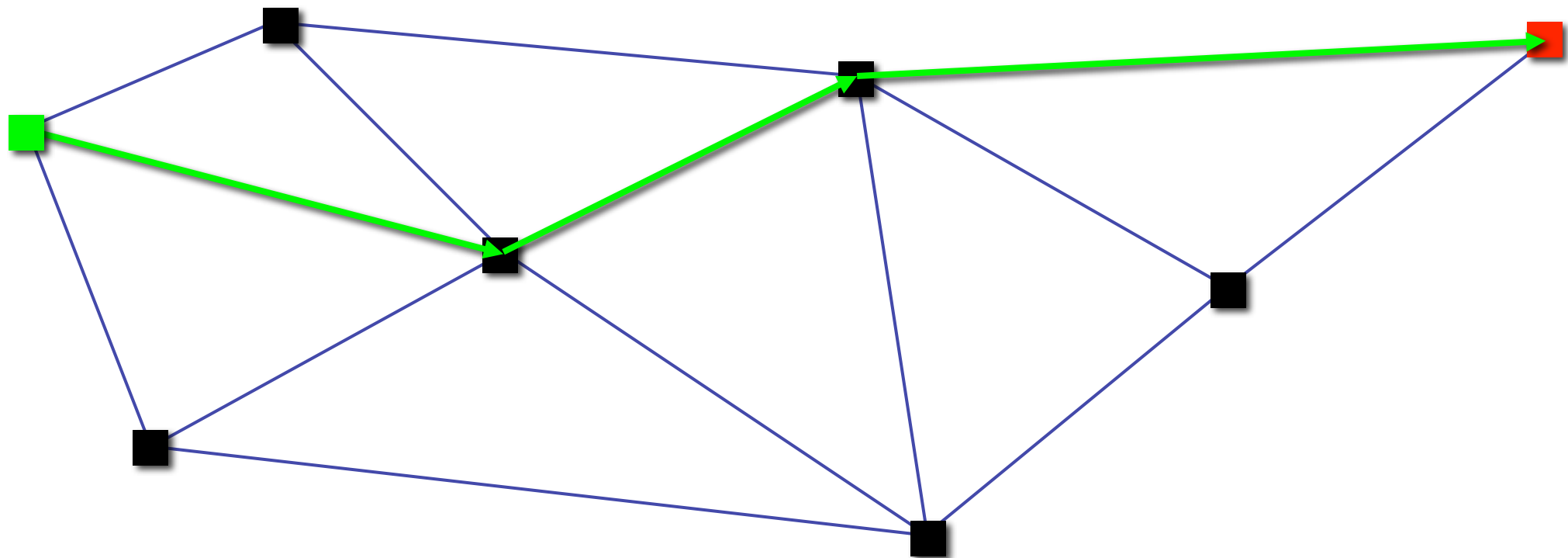
Method A

“Always go in the direction that takes you most directly towards Oxford.”



Method A

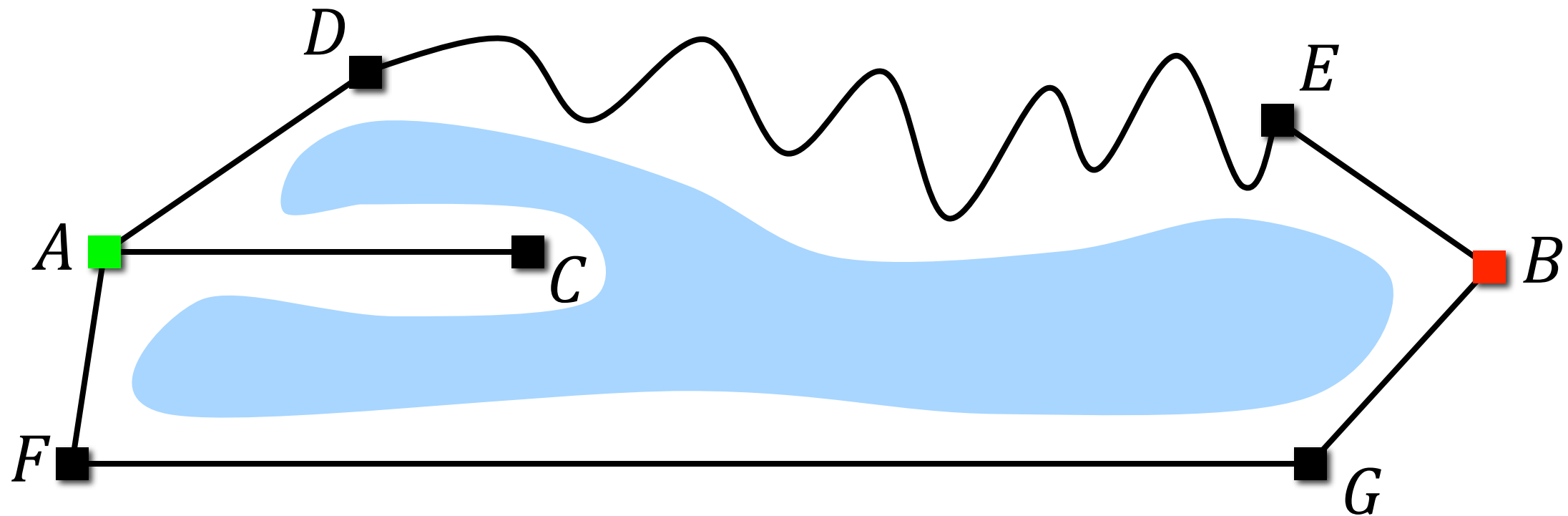
“Always go in the direction that takes you most directly towards Oxford.”



How good is it?

Method A

- *almost always* finds a path from start to finish.
- *sometimes* finds a path that not the shortest.



The programmer's dilemma

Any method that ...

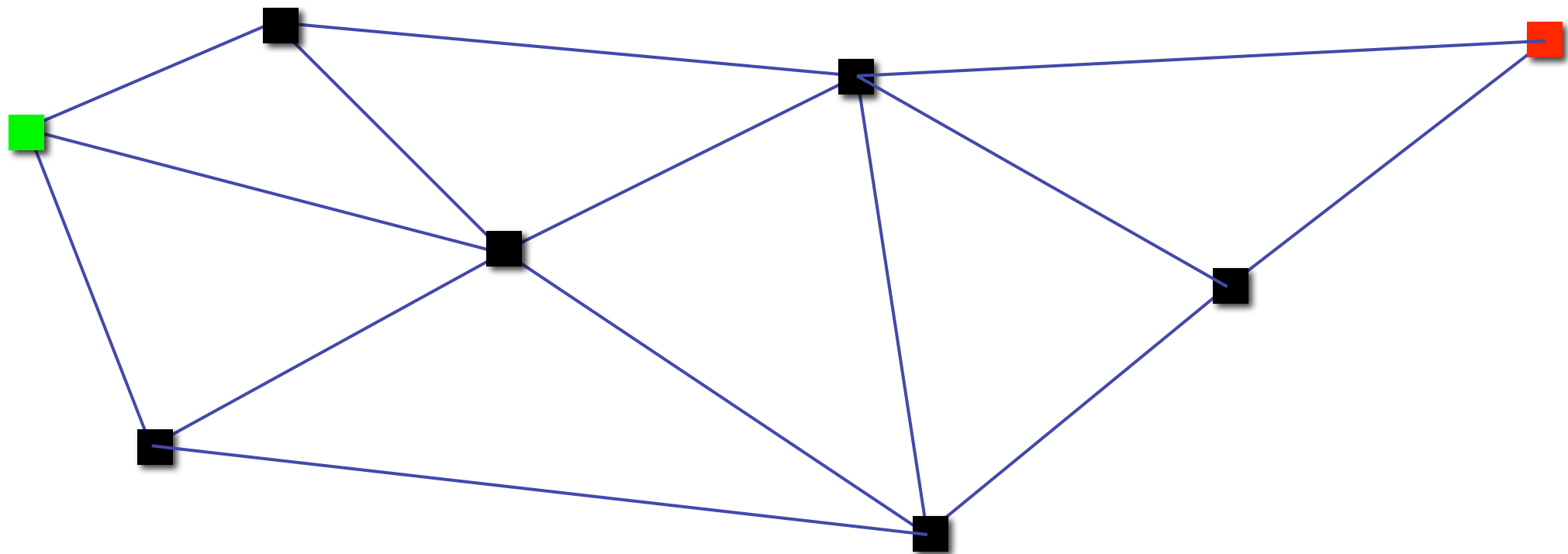
- usually works well but sometimes fails, or
- needs to be used with common sense

... is useless as a computer program.

Method B

Computers are very fast, so

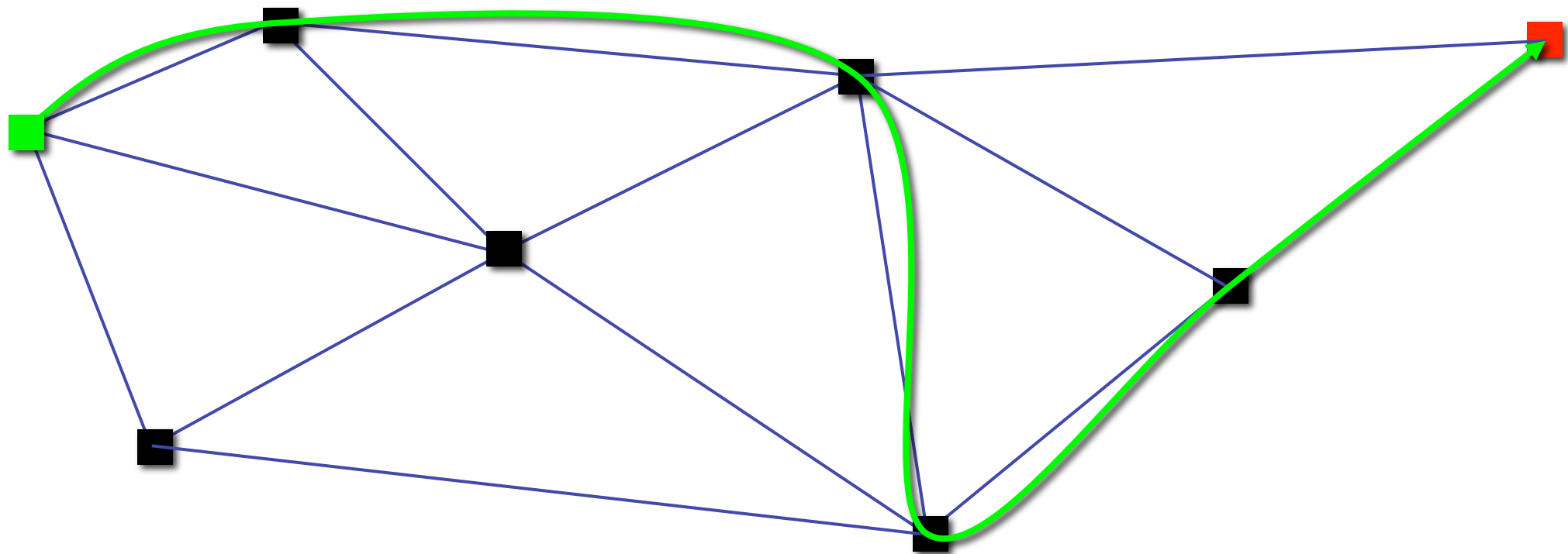
“Consider all routes from Manchester to Oxford, and choose the shortest.”



Method B

Computers are very fast, so

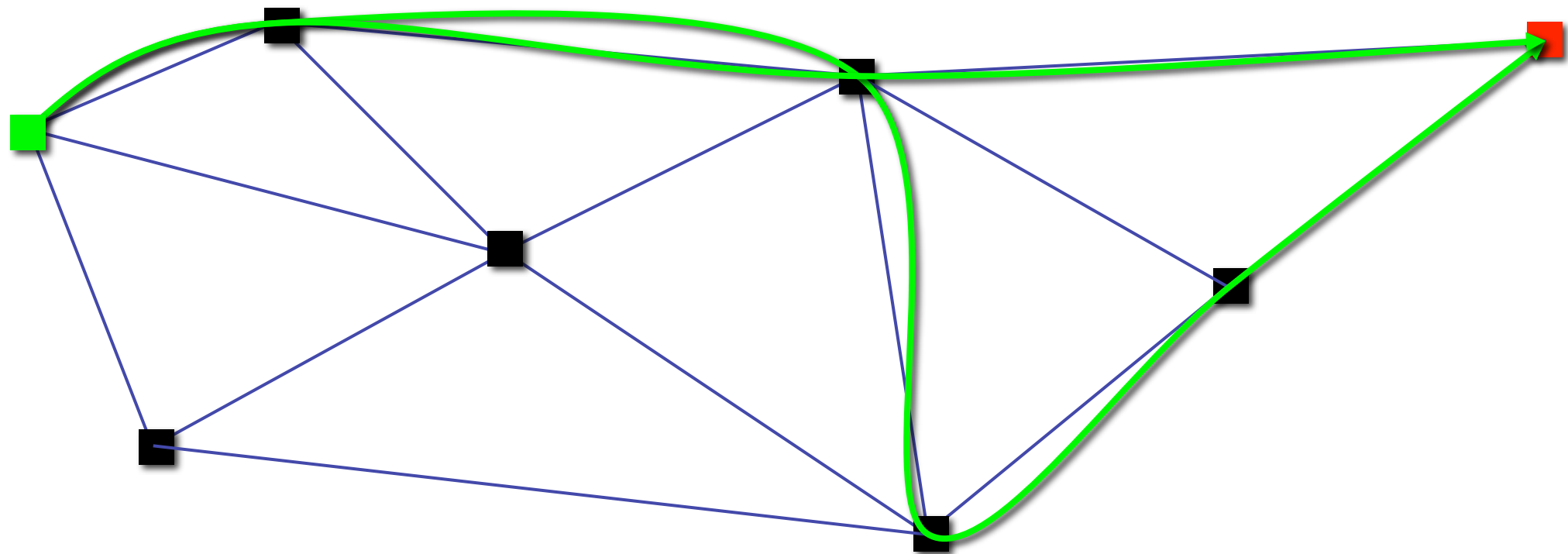
“Consider all routes from Manchester to Oxford, and choose the shortest.”



Method B

Computers are very fast, so

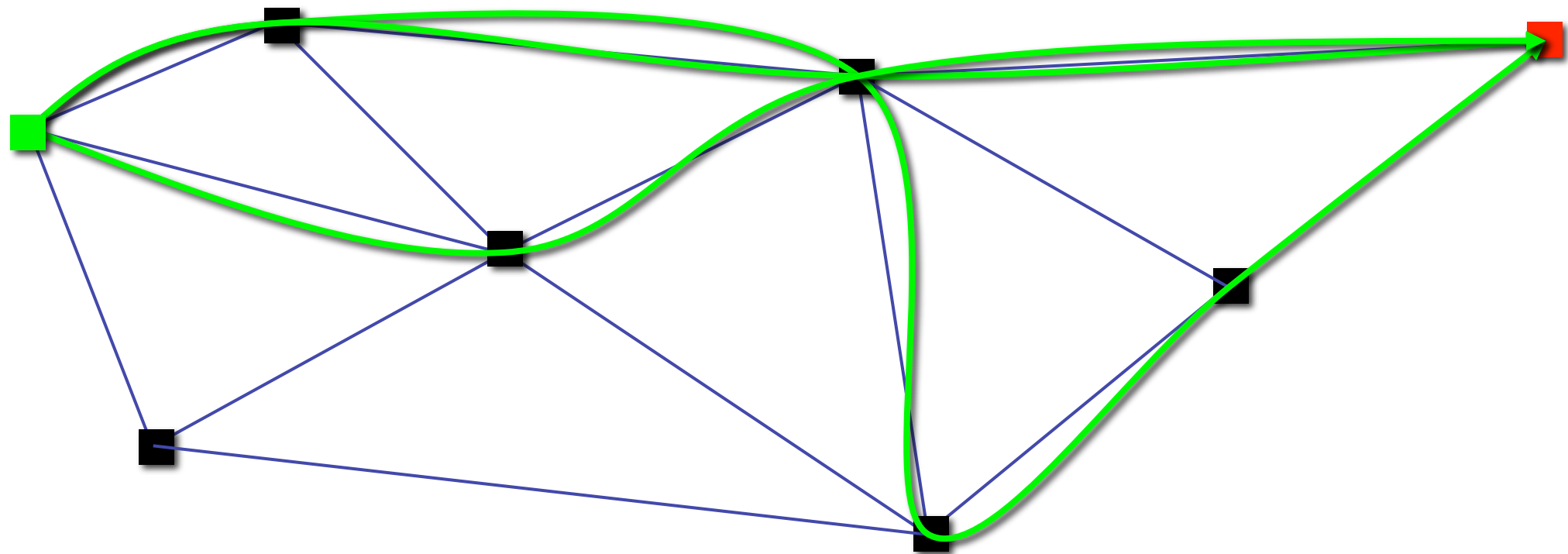
“Consider all routes from Manchester to Oxford, and choose the shortest.”



Method B

Computers are very fast, so

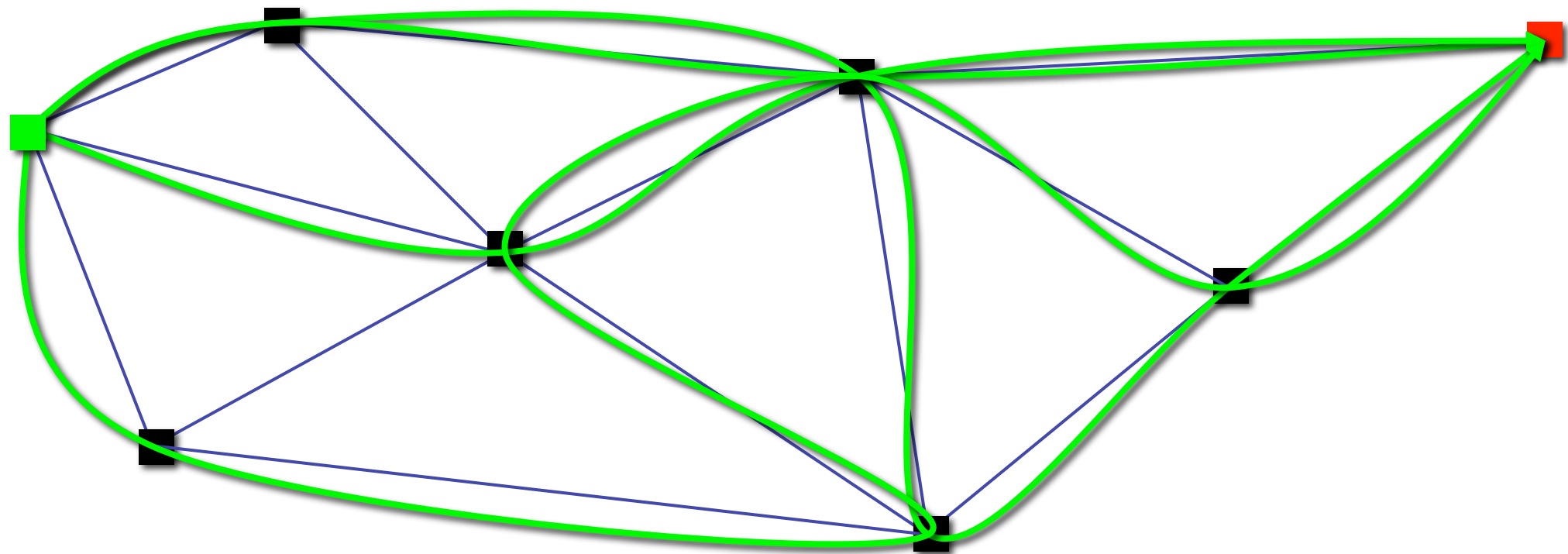
“Consider all routes from Manchester to Oxford, and choose the shortest.”



Method B

Computers are very fast, so

“Consider all routes from Manchester to Oxford, and choose the shortest.”



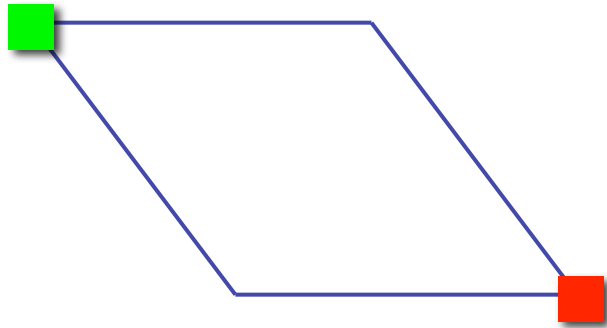
How good is it?

Method B

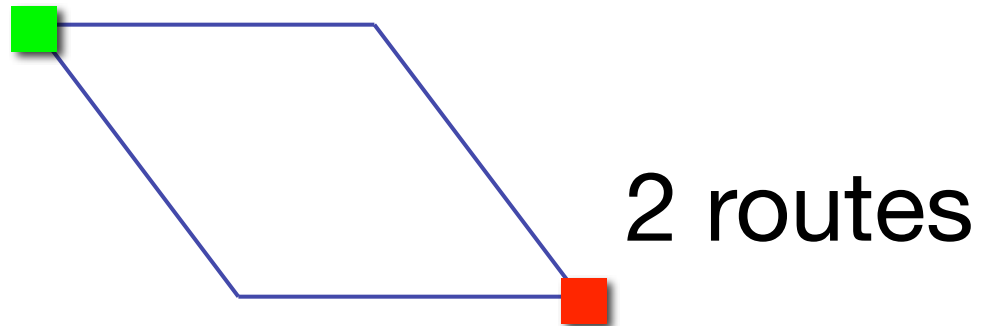
- will certainly find the shortest route.
- but the time taken grows too quickly as the map gets bigger.

How many routes might there be?

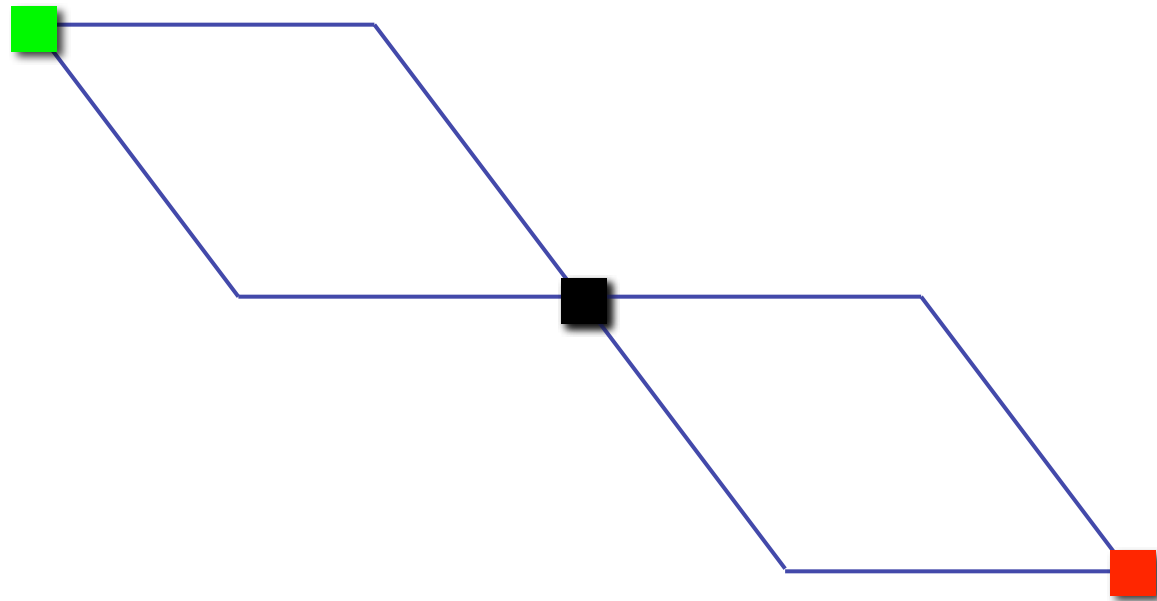
A growing number of routes



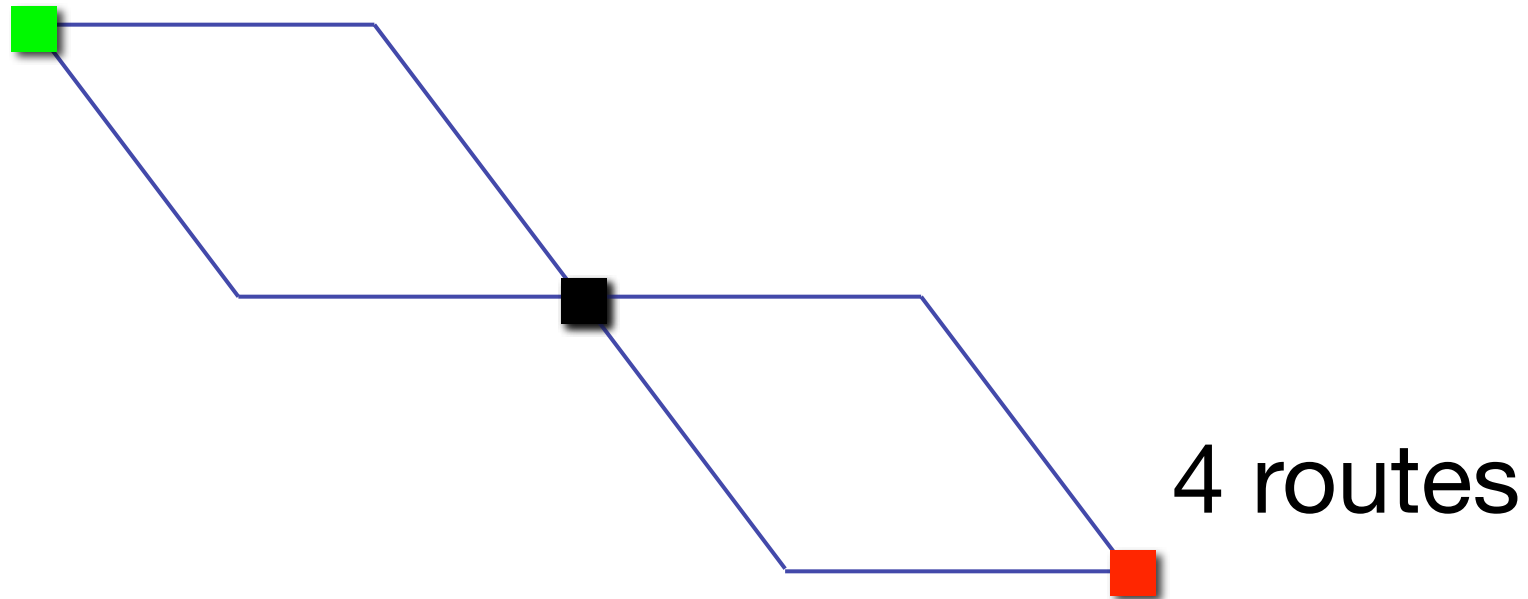
A growing number of routes



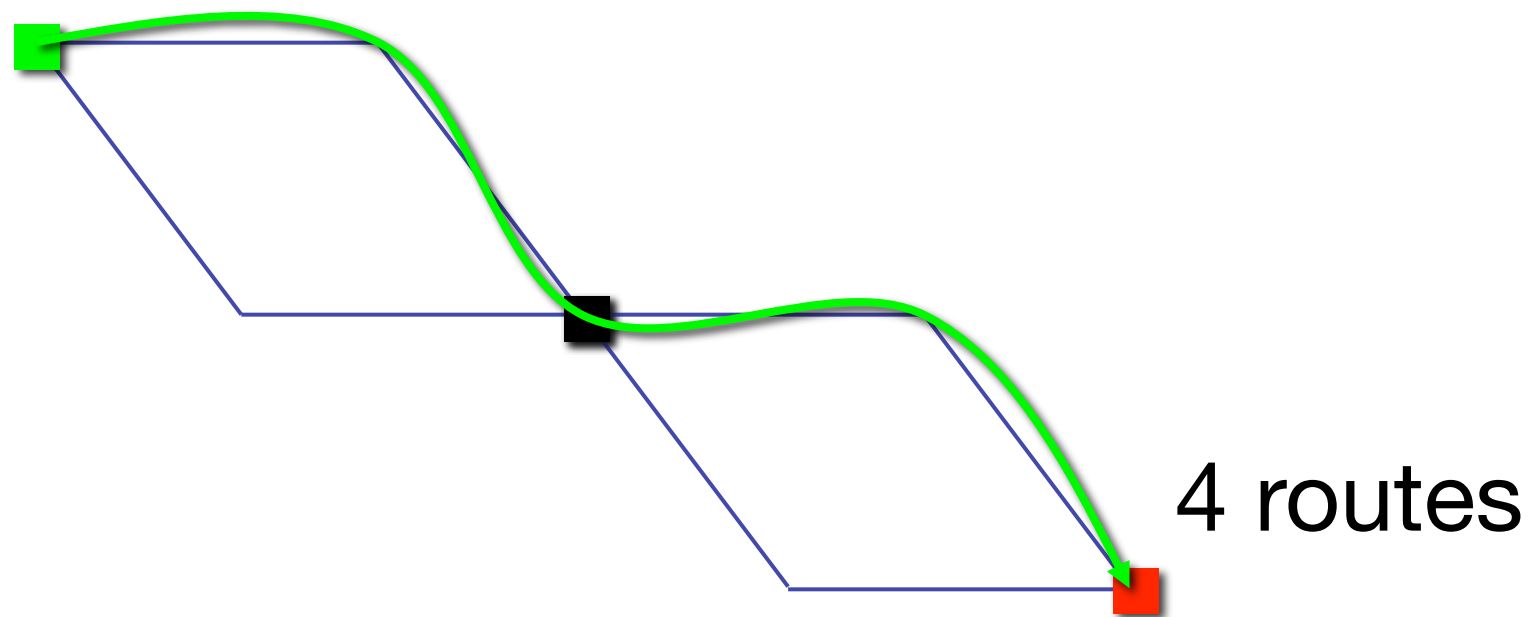
A growing number of routes



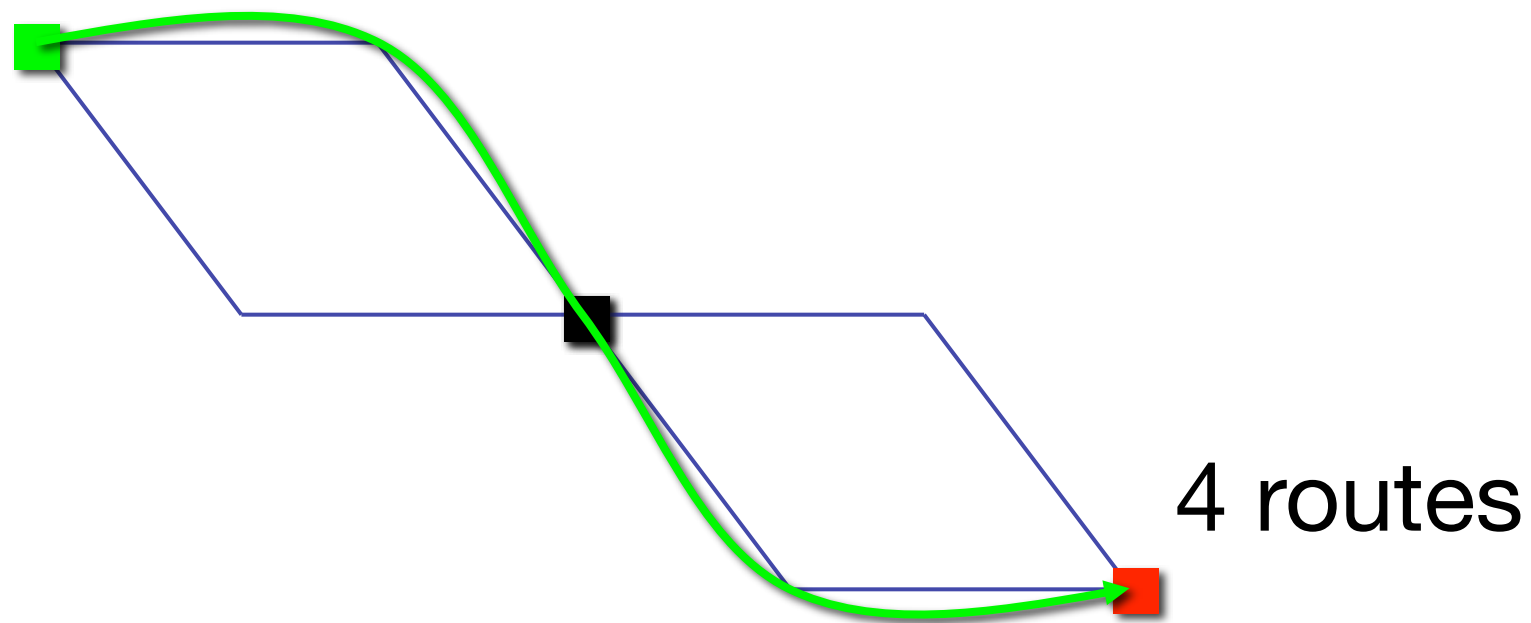
A growing number of routes



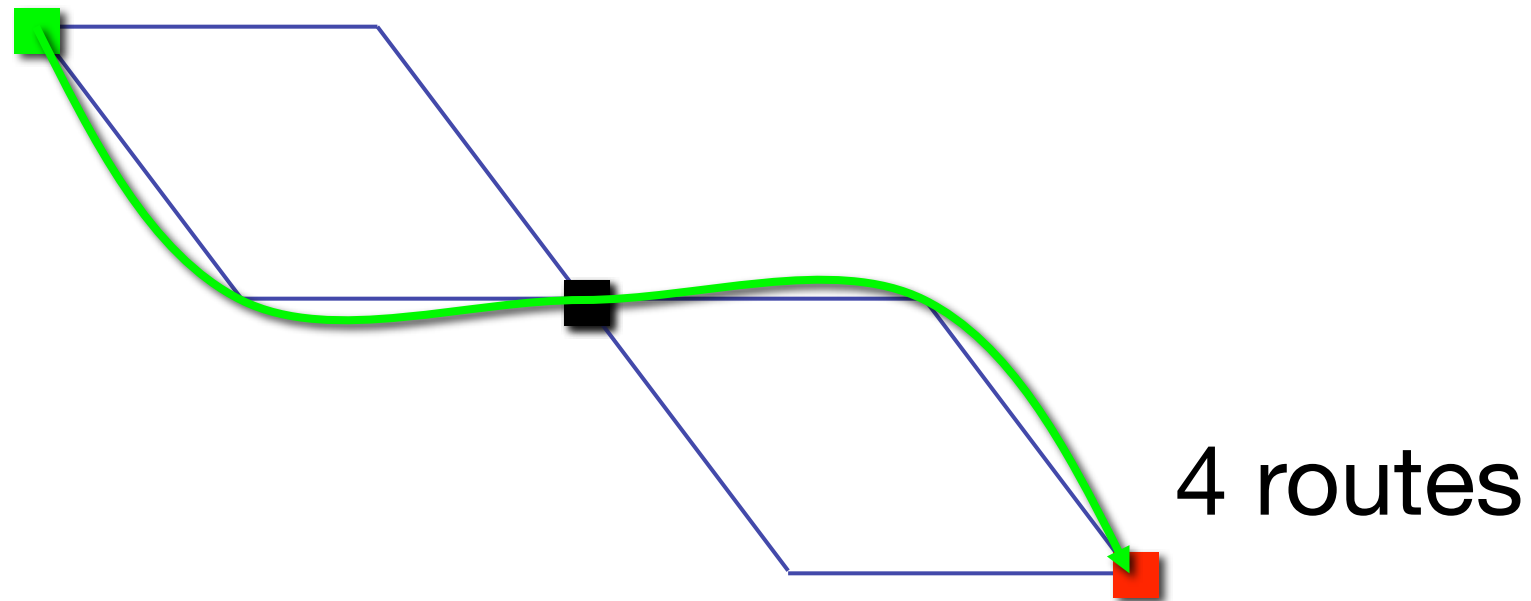
A growing number of routes



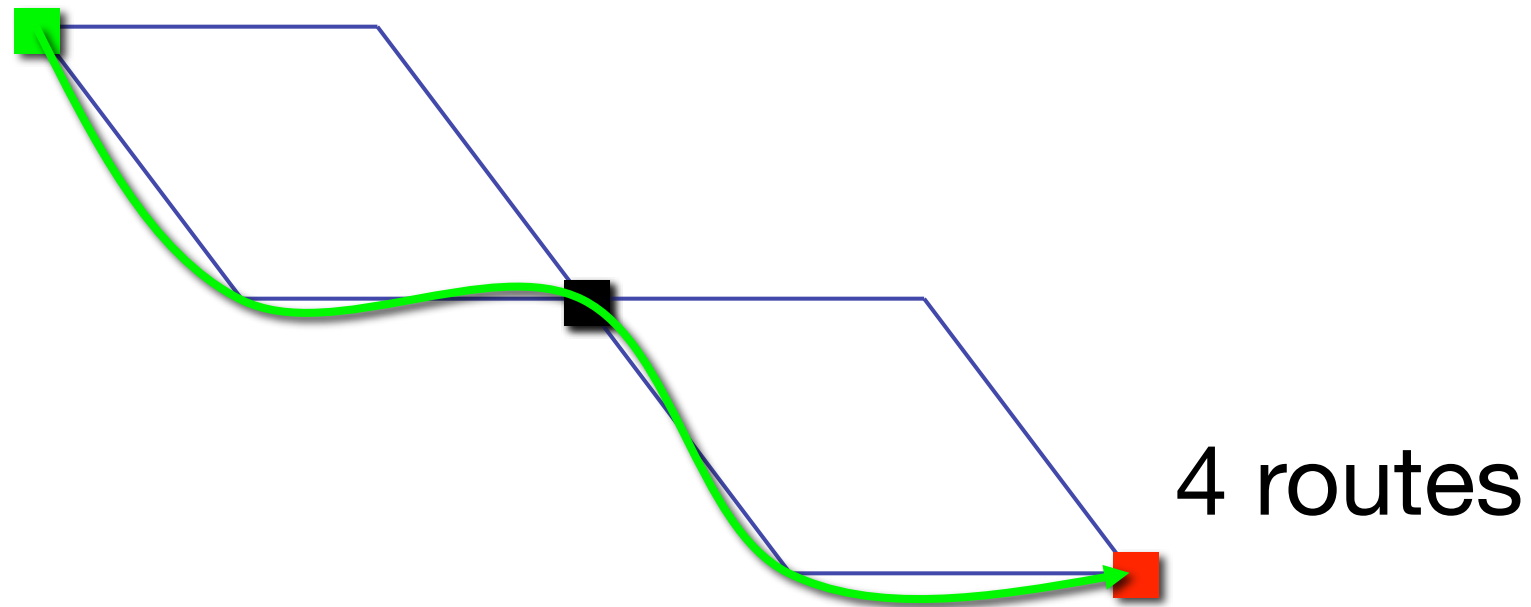
A growing number of routes



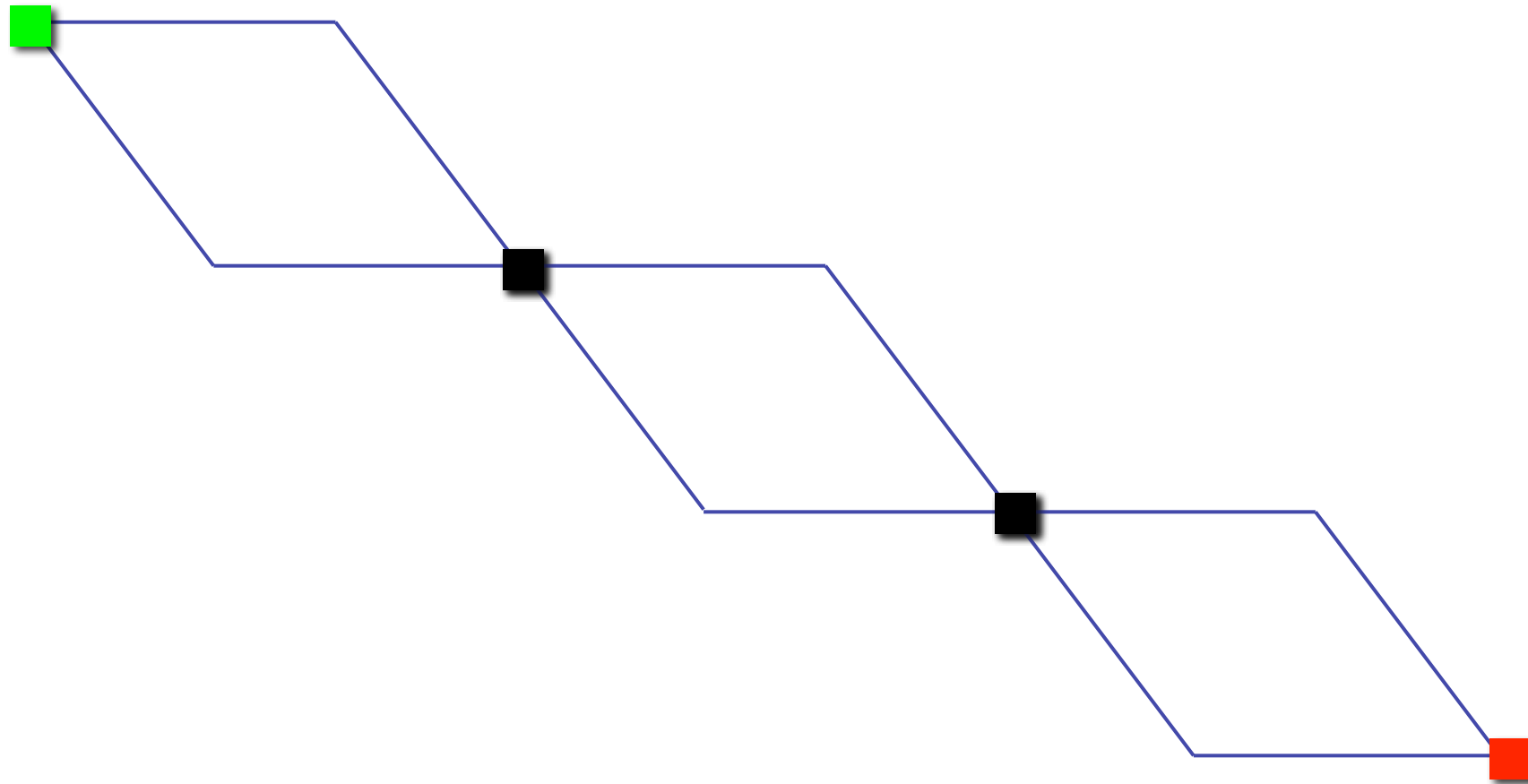
A growing number of routes



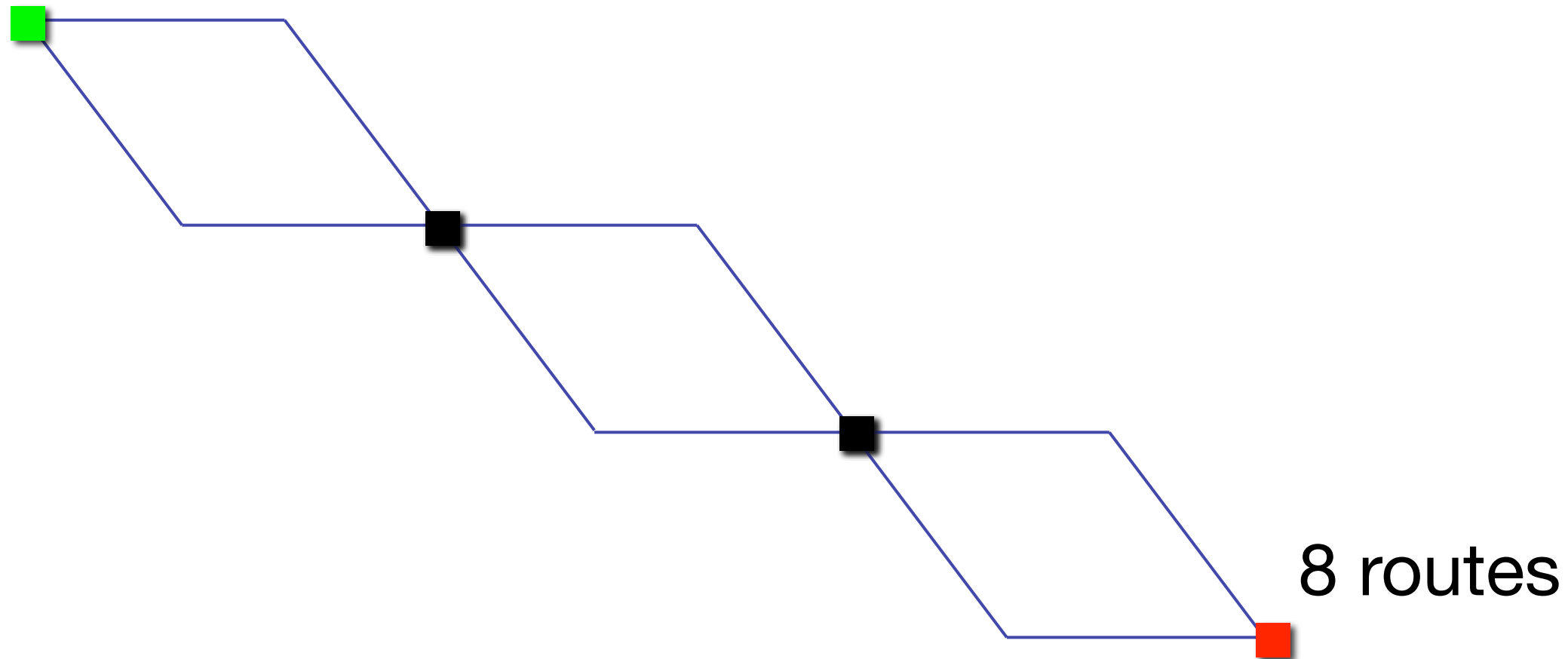
A growing number of routes



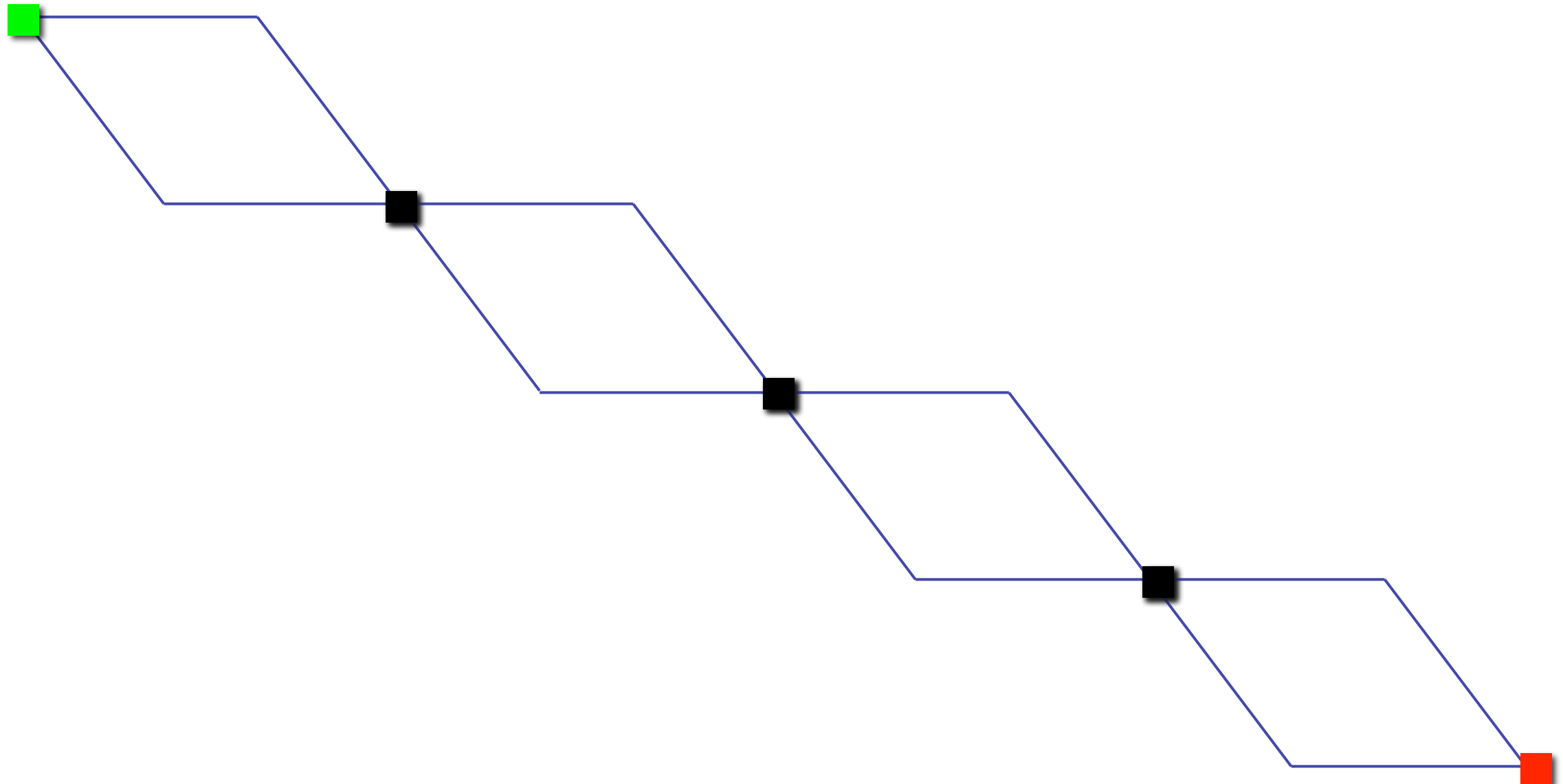
A growing number of routes



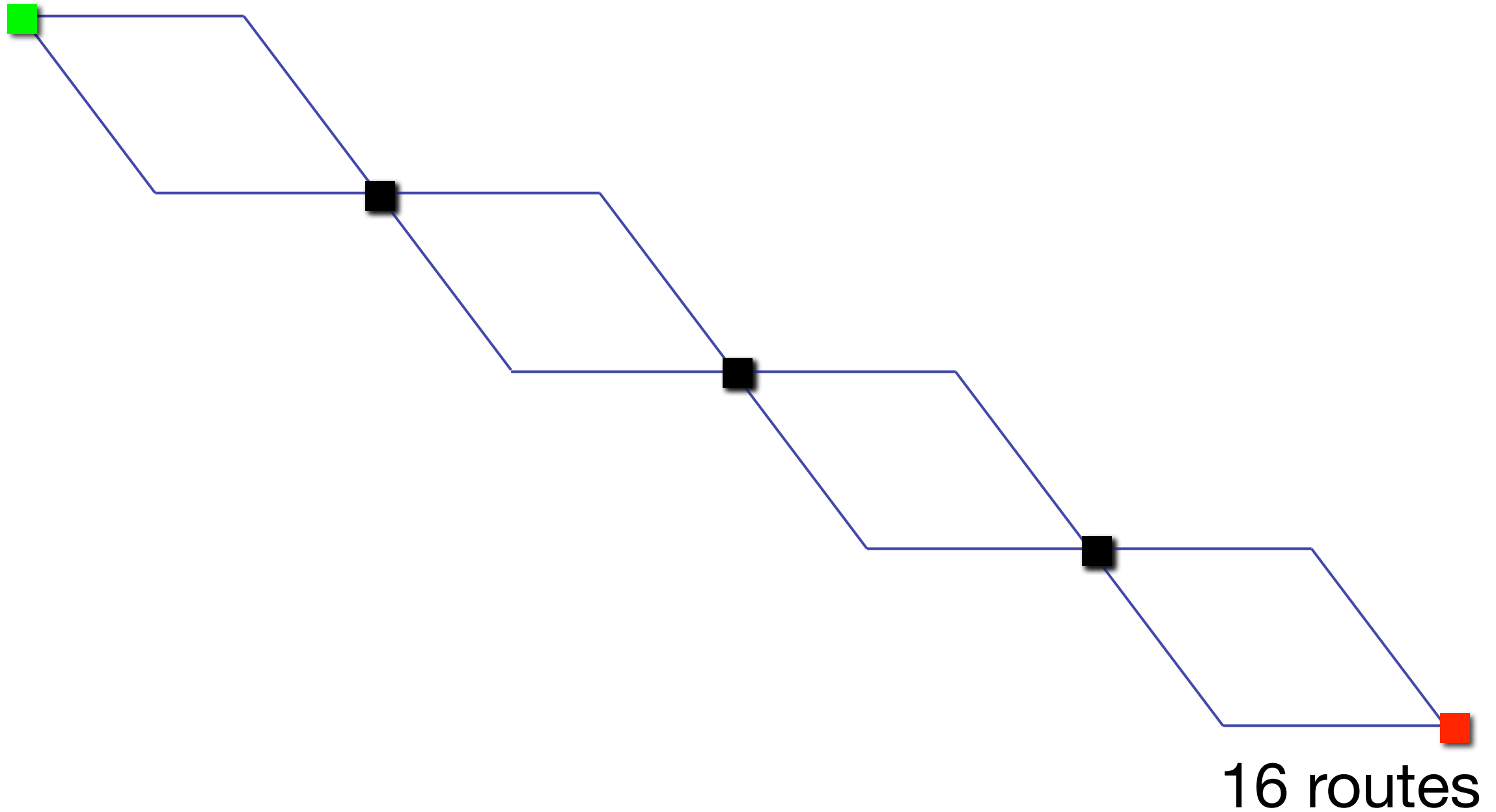
A growing number of routes



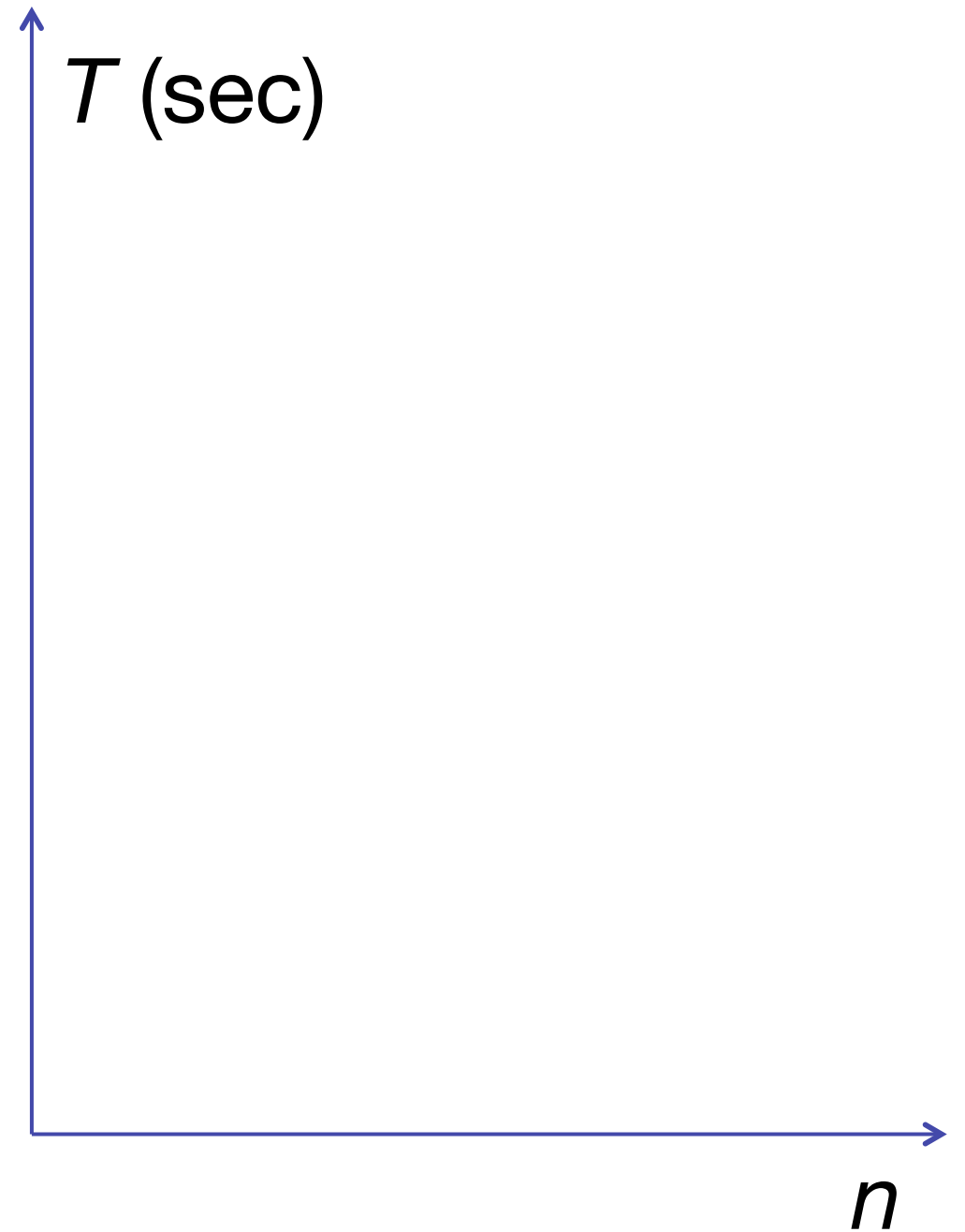
A growing number of routes



A growing number of routes

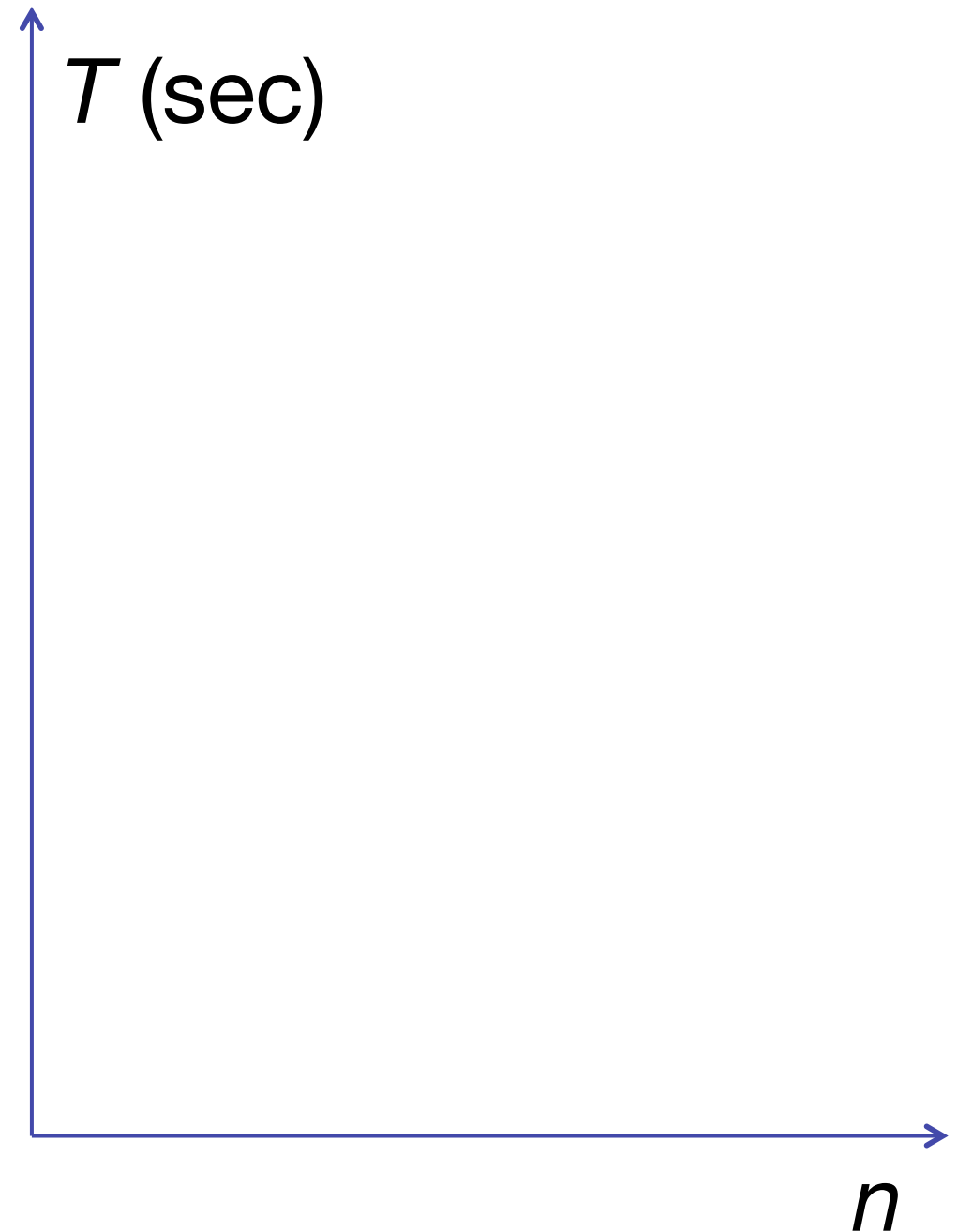


Starts slow and gets slower



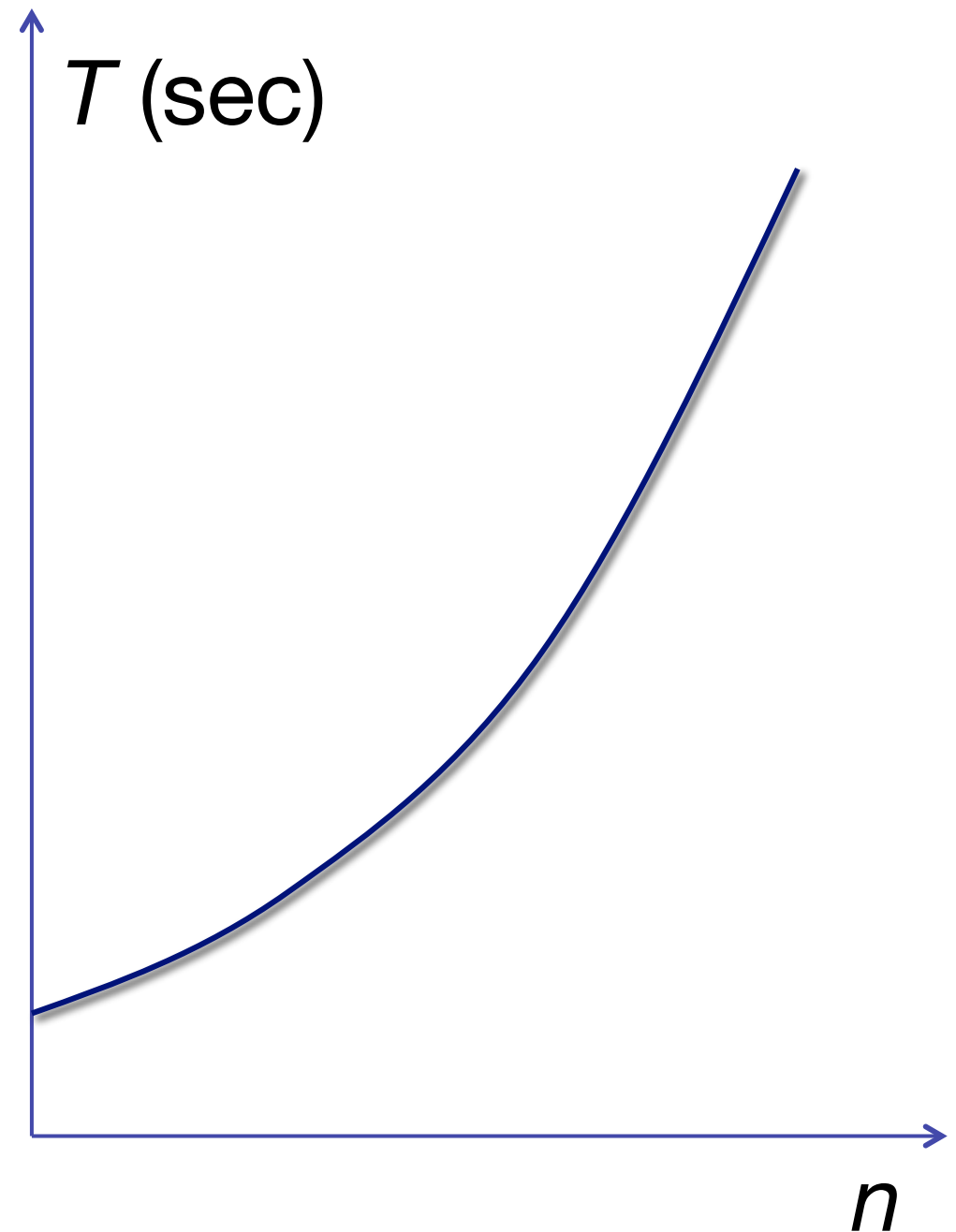
Starts slow and gets slower

The running time is proportional to $2^{n/3}$, where n is the number of towns.



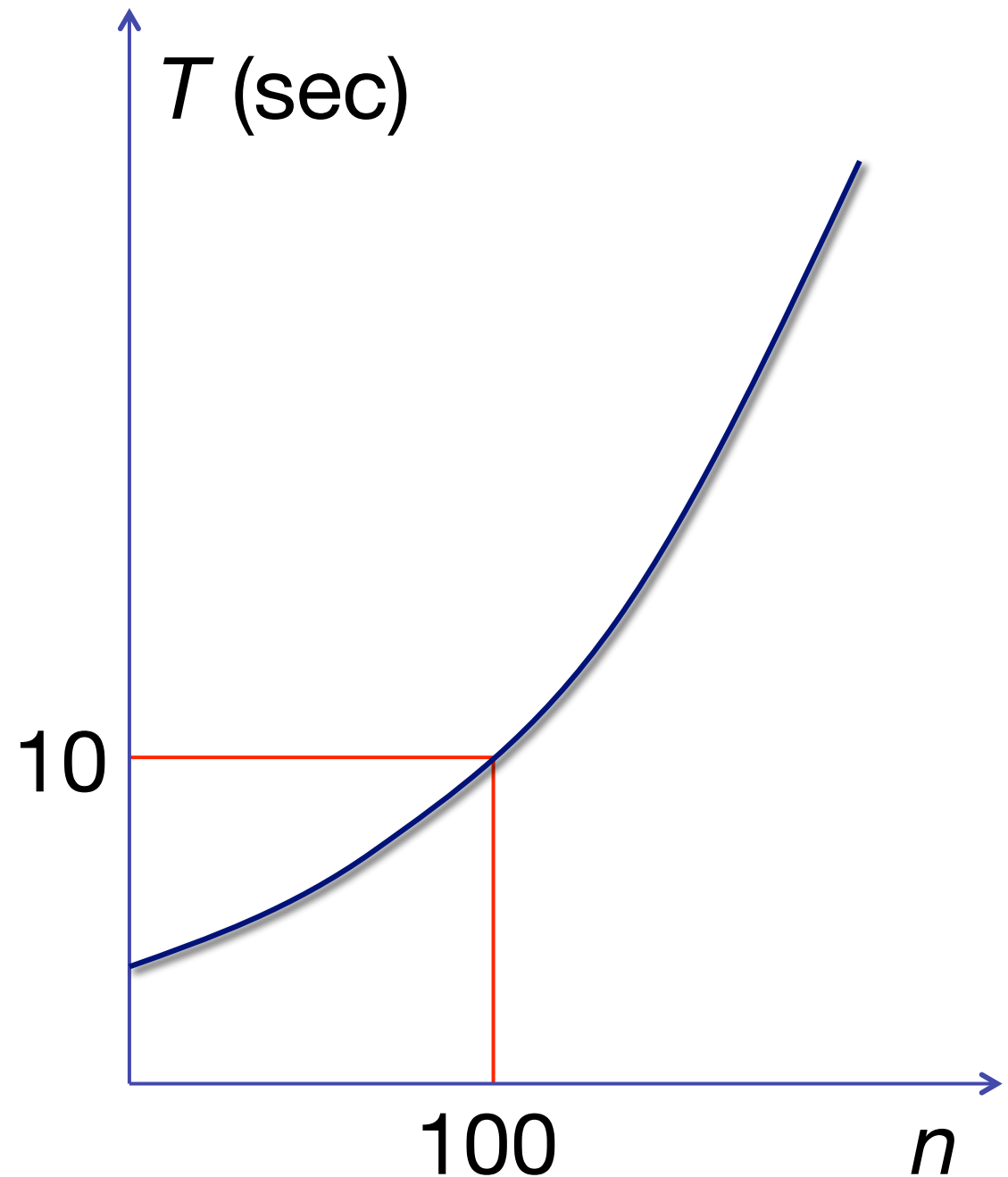
Starts slow and gets slower

The running time is proportional to $2^{n/3}$, where n is the number of towns.



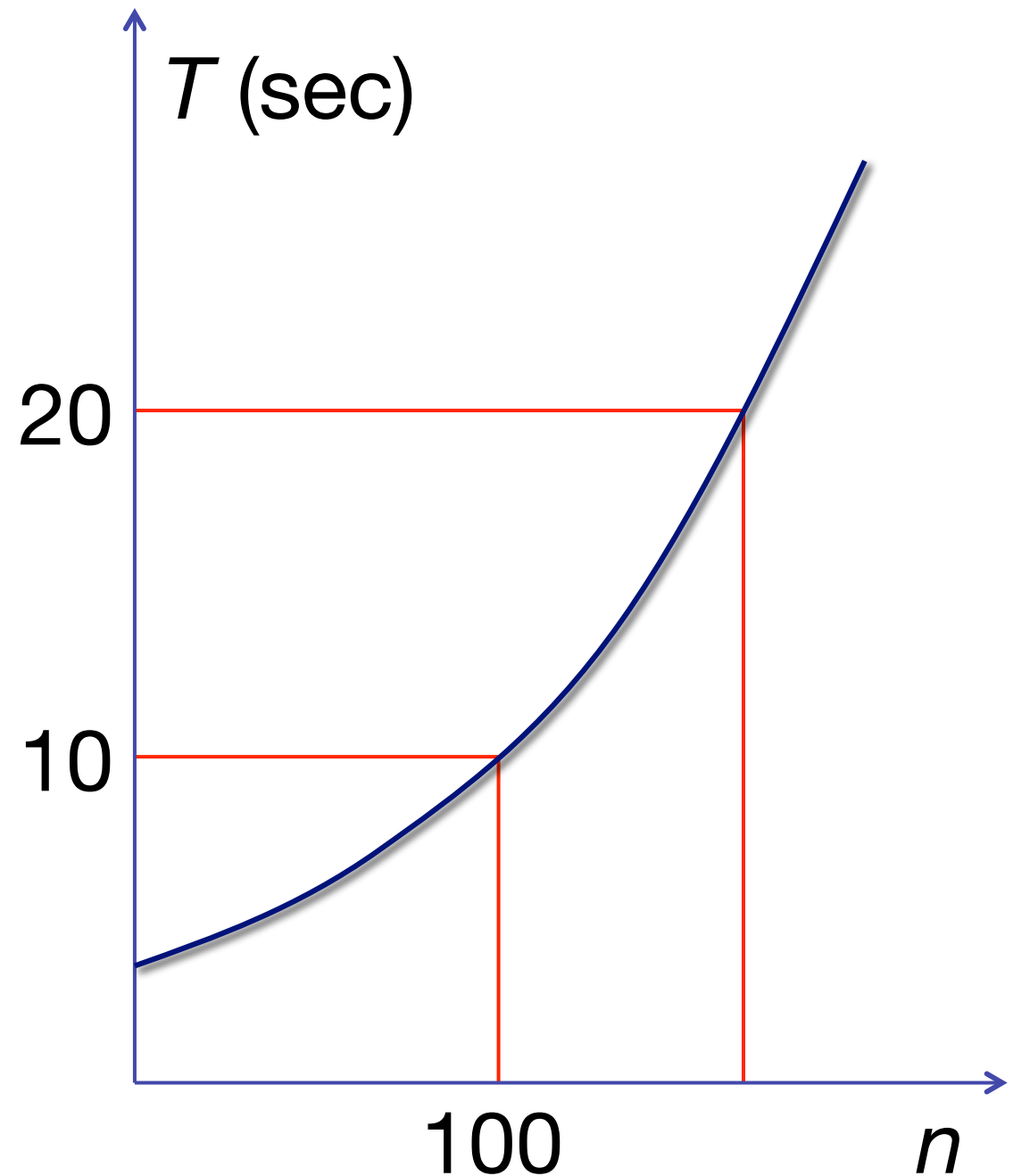
Starts slow and gets slower

The running time is proportional to $2^{n/3}$, where n is the number of towns.



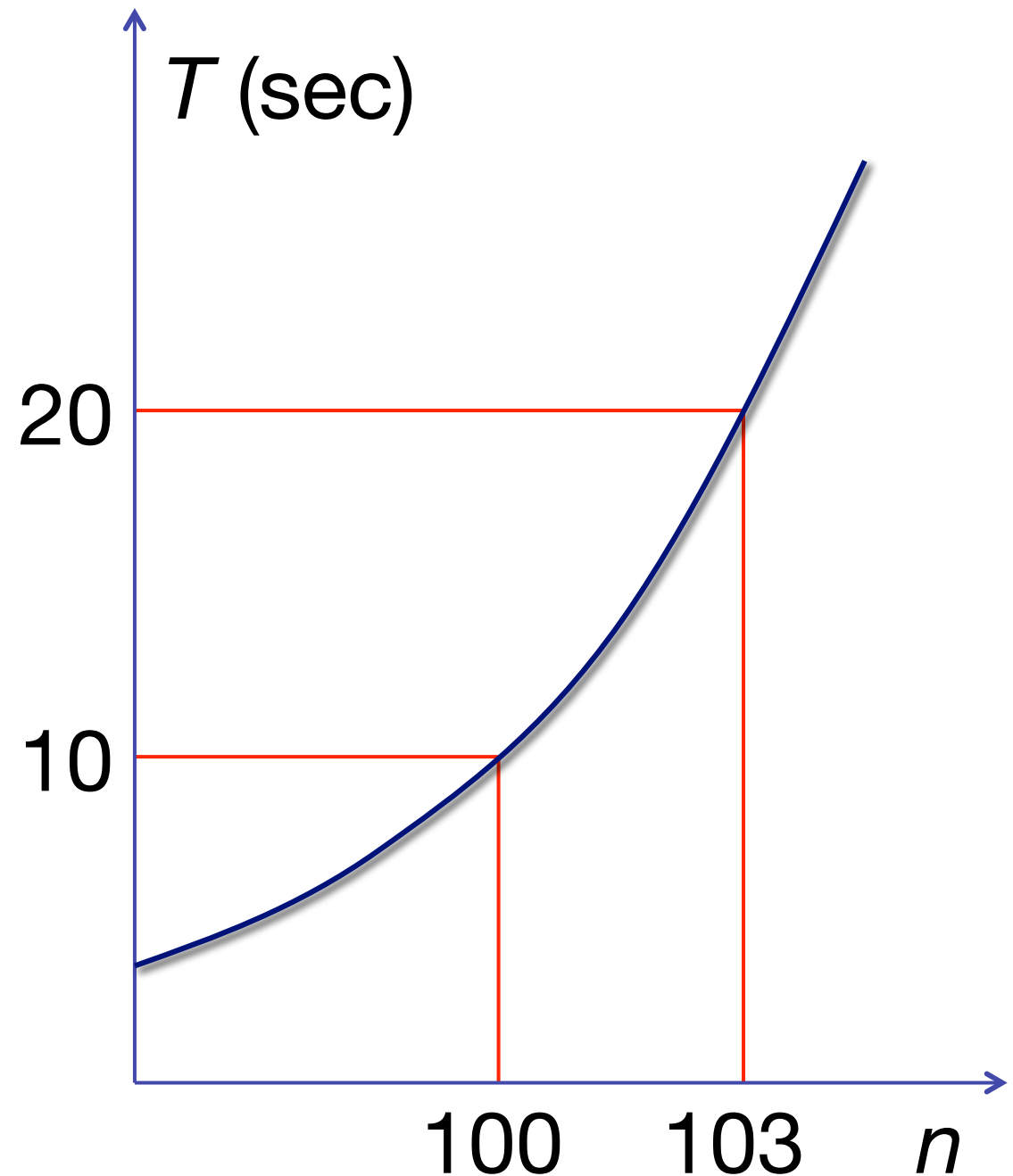
Starts slow and gets slower

The running time is proportional to $2^{n/3}$, where n is the number of towns.



Starts slow and gets slower

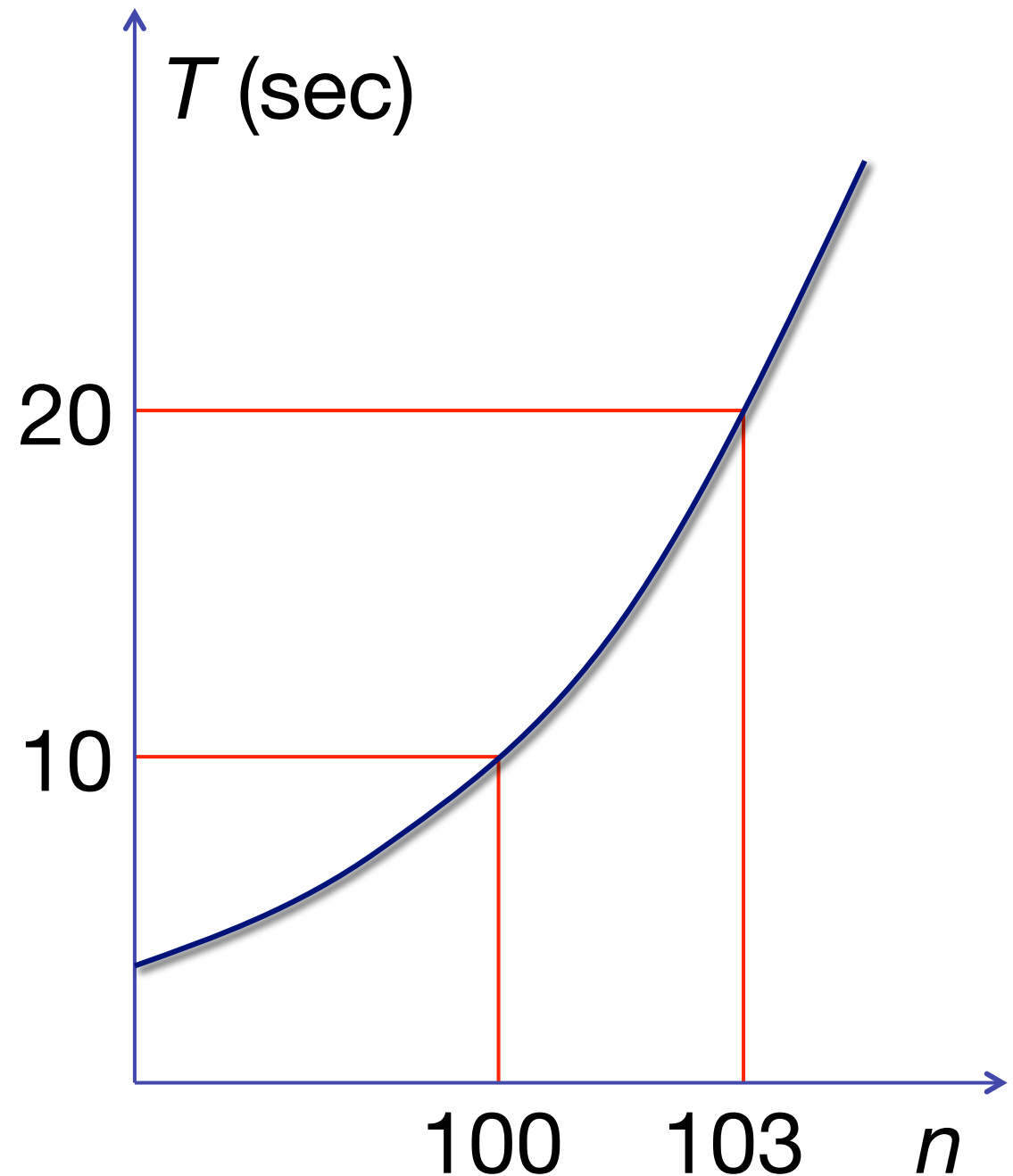
The running time is proportional to $2^{n/3}$, where n is the number of towns.



Starts slow and gets slower

The running time is proportional to $2^{n/3}$, where n is the number of towns.

- adding three more towns will double the time to find a solution.

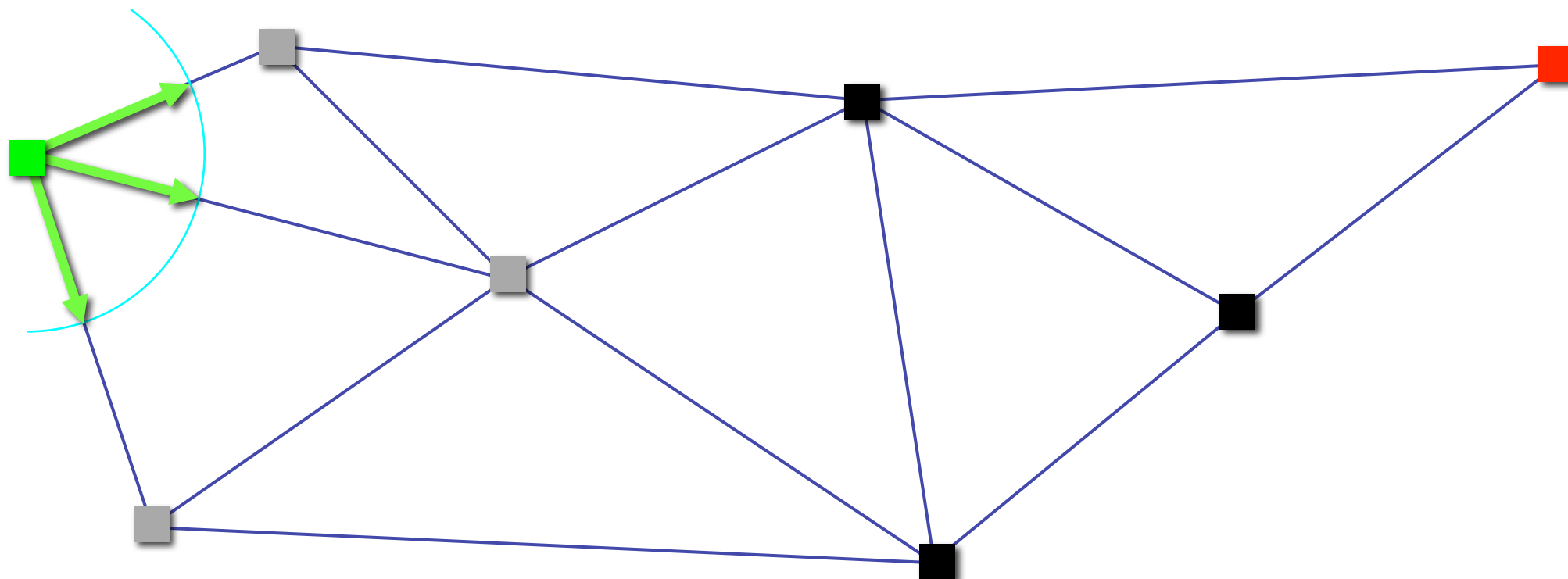


Method C

- Start an army of ants marching on each road out of Manchester.
- Whenever an army captures a new town, send new battalions on each outgoing road.
- The first ants to reach Oxford have gone by the shortest route.

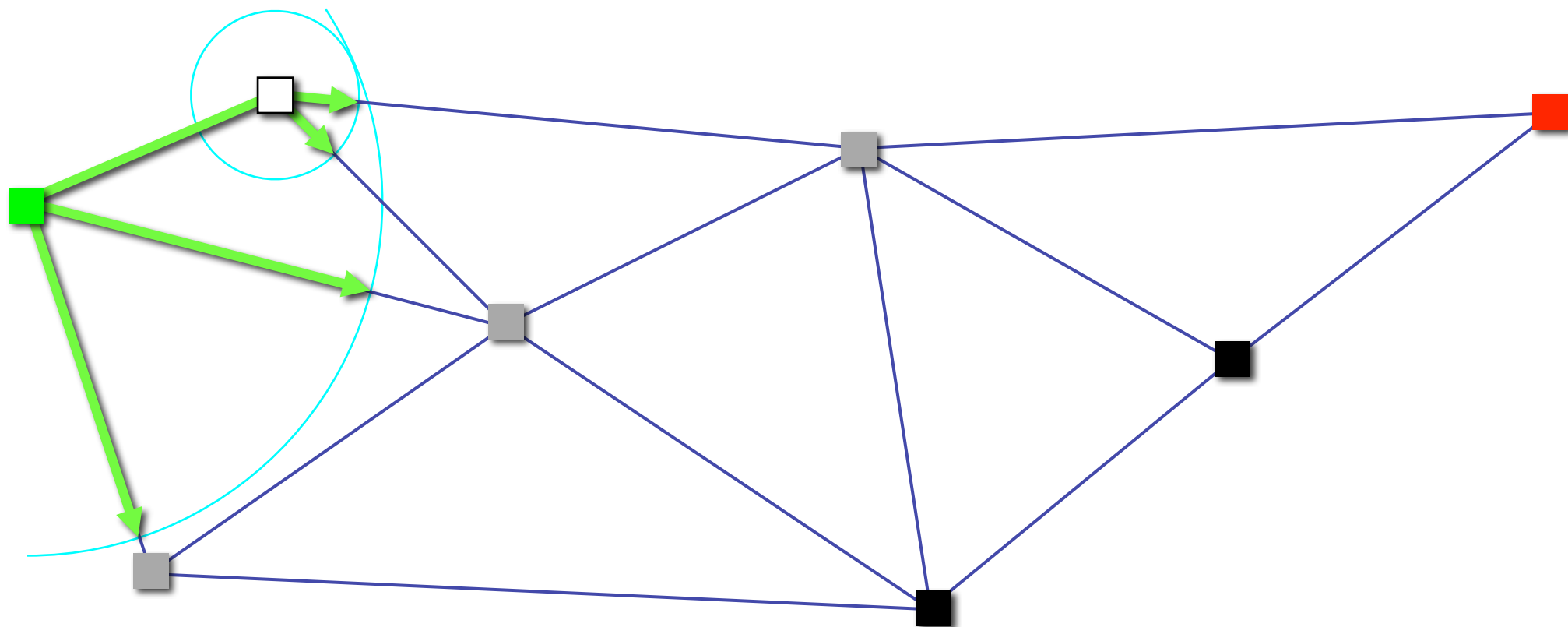
Method C

The ants start out from Manchester, all going at the same speed.



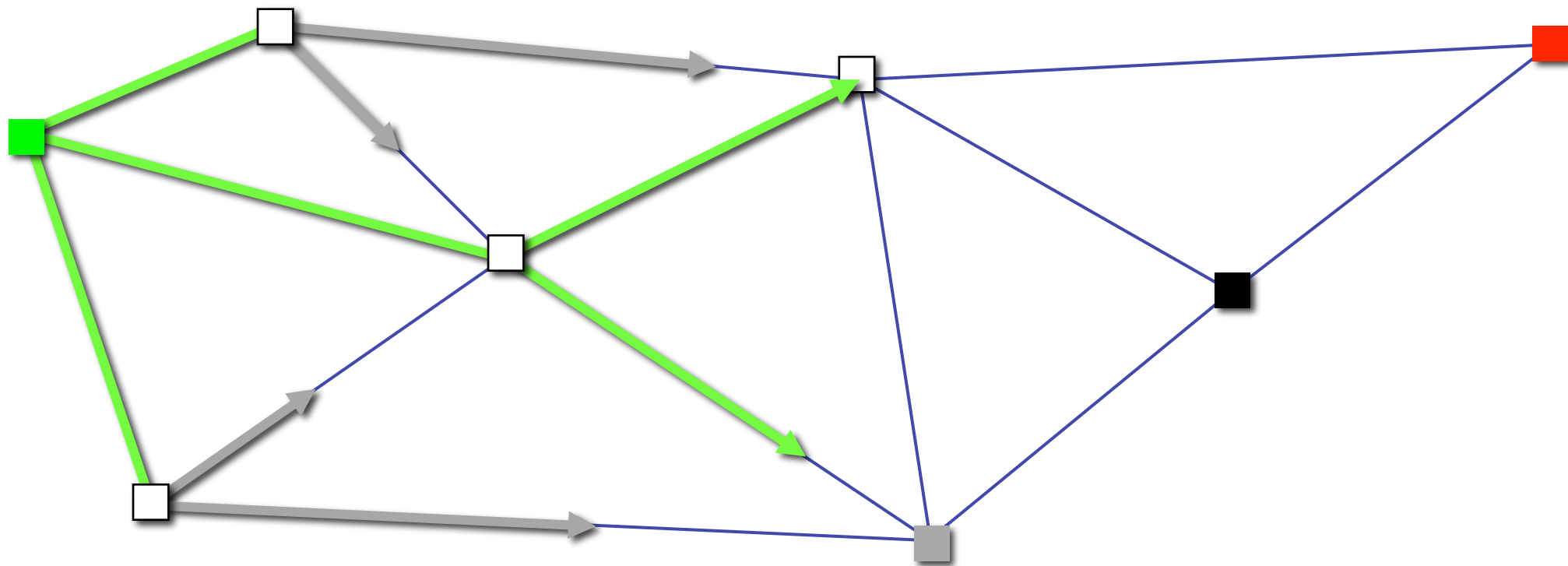
Method C

Each town that is captured becomes a new centre for expansion.



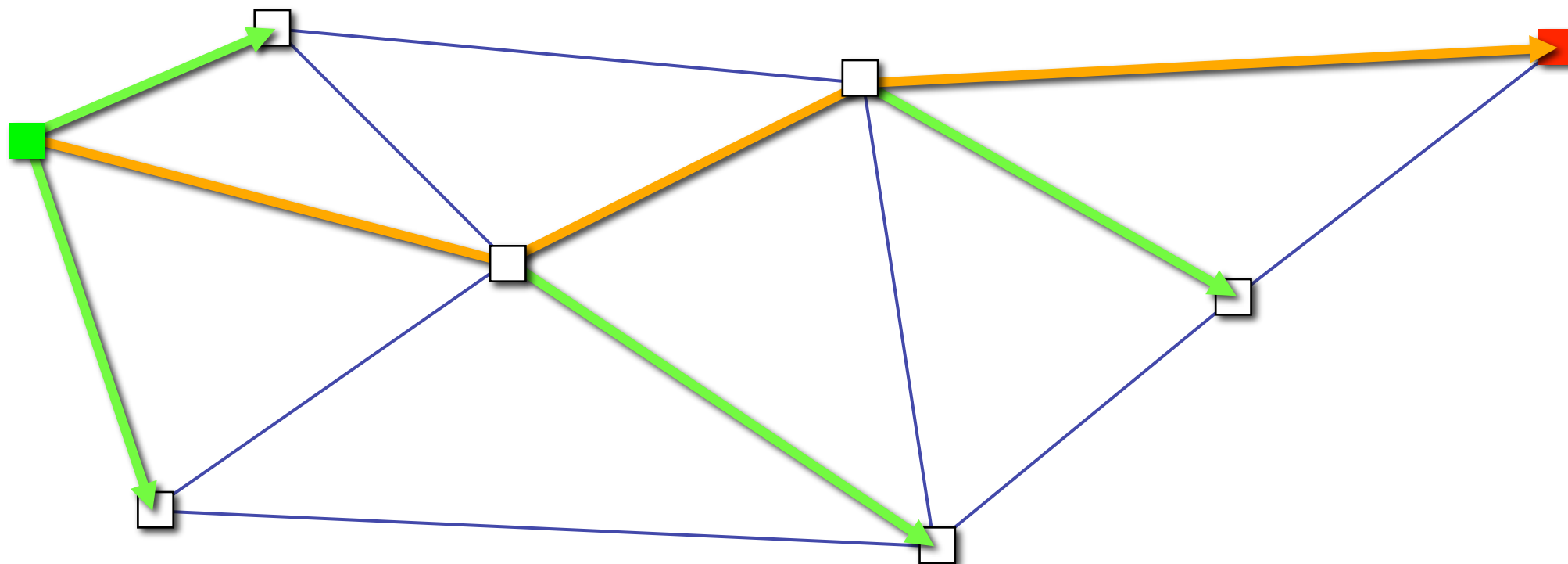
Method C

An army is disbanded if its goal will already be captured before it arrives.



Method C

When the first army reaches Oxford, we have found the shortest route.



Simulating ants by computer

- **White towns:** already captured.
 - *We already know the best route to get there.*
- **Grey towns:** under attack.
 - *Keep track of which army will get there first.*
- **Black towns:** not yet on front line.

Why the method is useful

Method C (Dijkstra's algorithm)

- Always gives the right answer.
- Does so in a predictable time.
- The time taken grows only gradually as the map gets bigger.

[Demo]

Writing it as a program

Writing it as a program

```
while (dst.colour != WHITE)
{
    t = ChooseMin();
    t.colour = WHITE;
    UpdateEstimates(t);
}
```

Java

Writing it as a program

```
while (dst.colour != WHITE)
{
    t = ChooseMin();
    t.colour = WHITE;
    UpdateEstimates(t);
}
```

Java

```
while dst.colour  $\neq$  white do
    t := ChooseMin( );
    t.colour := white;
    UpdateEstimates(t)
end
```

Oberon Pascal



Computer science

- It's *not* about learning new programming languages.
- It *is* about understanding why programs work, and how to design them.